Fig. 7 is a drop table illustrating superpixel definition for last-stage expression of a four-bit error-diffusion system;

Fig. 8 is a like table but showing superpixel definition in so-called "superpixel families", for four different permutations (identified as "0" through "3") of a four-bit system;

Fig. 9 is a diagram illustrating use of a so-called "expansion matrix" for conversion from error-diffusion state to superpixel assignment (in an example converting from 12 dots/mm and three bits, to 25 dots/mm and two bits);

Fig. 10 is a table illustrating a four-permutation superpixel definition (at 25x25 dots/mm);

Fig. 11 is a group of four coordinated graphs, of which the first (upper) pair of graphs relates halftone value to "contone" (continuous tone) color tonal level, for a single-bit binary system and a two-bit system — and so illustrate a conceptual extrapolation of error diffusion from binary to multibit; and of which the second (lower) pair of graphs represents the contone functions themselves — i. e., illustrates application of linearization curves and thresholds to multilevel error diffusion;

Fig. 12 is a linearization curve for black — i. e., a graph of linearized black vs. contone input, nine and eight bits per pixel respectively — and particularly representing a preferred embodiment that is part of a commercial product;

Fig. 13 is a diagram like Fig. 4 but for the Fig. 1 ink-limiting and plane-split stages (particularly representing acquisition of the "factor" described in the associated text); and

SUBSTITUTE SHEET 6/14/5

Fig. 14 is a highly schematic diagram showing cyan (C) and magenta (M) separation in the Fig. 13 limiting and split stages.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

1. APPARATUS-MODULE AND BUSINESS-ENTITY INTERRELATIONS

Preferred apparatus embodiments of the invention involve three major modules 113, 121E, 141 (Fig. 1), one of which can include an optional internal module 121N. Of these four units, two are parts of the environment of the invention, not elements of the invention itself as most broadly regarded: a computer 113 and an internal RIP 121N.

The remaining two units are elements of at least some of the previously introduced major apparatus aspects of the invention, again as most broadly conceived. These are the printer 141 (excluding its internal RIP 121N) and the processor or external RIP 121E. In addition, <u>provision</u> of one or the other of these two units 141, 121E is an element of at least one of the major method aspects of the invention.

Essential to the objectives of any such system or method is existence of an image 111, which may be derived from a separate source and then pass through an entry mechanism 112 into the computer 113 (as suggested in Fig. 1). There an image is most typically subject to modification in a general-purpose microprocessor 114, 119E that

SUBSTITUTE SHEET 6/14/5

permutation one-quarter of the time — or to use them in different proportions.

Finally, it is possible to choose a small matrix or a large one. A larger matrix will show less patterning, but require more system memory. Fig. 9 provides an example of how it all works together, when the above superpixel definition is applied.

As noted earlier, it remains to document the 25 dot/mm to 25 dot/mm superpixel family (Fig. 10). It can be considered an identity, and is uninteresting.

This time the application goes from a 25x25 dot/mm cell to a 25x25 dot/mm cell. The present inventors advise against use of superpixel families that average a nonintegral number of drops, as increased granularity results.

7. HALFTONING

The stage that feeds superpixeling is the halftoning algorithm. A preferred algorithm for use with the present invention is error diffusion.

Error diffusion is very well known in this field. It was originally conceived as a way to transform data from multibit to binary (that is, single-bit). As an example consider an area fill, defined at 25 dots/mm, 8 bits per pixel. The whole area has the same value: tonal level 130 (in a conventional scale from zero through 255).

The only available choice is between firing a drop on a given pixel location or not firing it. If the input value is 0, then the system refrains from firing (0). If instead the input value is 255, then the system fires (1).

If the input value is somewhere in between, then the system goes to the closest point, but it has committed an

SUBSTITUTE SHEET 6/14/5

error; therefore it must try to commit the error in the inverse sense when moving to the neighboring pixels.

In the example, tonal value for the first pixel is 130. This is closer to 255 than to 0, so the system decides to fire (1). It has committed an error of +125, that it must then distribute among the neighbor pixels.

Assume that the next pixel receives a fourth part of the error of the previous pixel (that is, -31 counts). Then, the system must calculate that the second pixel has a value of 130 - 31 = 99. This total input value of 99 is closer to 0, so the system decides not to fire (0) — but thereby it commits an error of -99, that in turn it must propagate to the surrounding pixels (some of which will also receive error from the first pixel). This process proceeds through hundreds of thousands, or millions, of iterations to complete an image.

To fit this algorithm into the present invention, a few modifications are required. These are explored in the two subsections below.

(a) <u>Multilevel error diffusion: thresholds</u> — A first step is to conceive of a way to implement the binary outcome of classical error diffusion into a multievent (<u>i. e.</u> multibit) outcome. That is, it is no longer a binary decision between firing or not firing a drop, but rather which superpixel family to choose.

If the system is halftoning at 25 dots/mm, two bits, we'll have four superpixel families to choose among. The concept must be scalable to 12 dots/mm at four bits (sixteen superpixel families) — and even further, to six dots/mm, four bits.

Fig. 11 shows (in the two upper graphs) how the error diffusion algorithm can be expanded from binary to multi-

SUBSTITUTE SHEET 6/14/5

bit. At the same time, the output value has been decoupled from the actual number of drops being fired.

The graphs show how the contone input can be divided into a number of regions equal to 2*n* - 1, corresponding to *n* bits per pixel at the output. Besides the two natural thresholds, which are 0 and 255, new thresholds appear: A and B.

Using this strategy, input values that are closer to A generate an output to superpixel ("SPX") family 01; those closer to B will be assigned to SPX 10, and so on. Errors propagate in the classical way described above.

This explanation is the real picture for a 2 bit/pixel output, easily expanded to 4 bit/pixel or whatever is required. Although Fig. 2 shows the ED thresholds A and B equally spaced from 0 and 255, because of linearization considerations this relationship is not maintained.

(b) *Linearization* — The classical ED algorithm was originally conceived for monitor screens. On a monitor screen each pixel is clearly bounded, and never overlaps with the surrounding pixels. These constraints facilitate good linear response of the algorithm.

In inkjet printing, however, the printed drops do overlap. The macroscopic result is, that error diffusion is no longer linear.

It is accordingly widely known in this field that a linearization file should be created. The linearization file is applied to the continuous-tone information in advance of ED processing (Fig. 11, lower graphs).

The composite of the two functions *linearization* and *error diffusion* is supposed to be the identity — so that a linear contone gradient still comes out linear, once halftoned. In addition, because the linearization curve may assign a single image tone to different consecutive

SUBSTITUTE SHEET 6/14/5

inputs and thereby create contouring, the linearization function also transforms the data from eight bits to nine. This transformation minimizes the contouring effect.

The graphs also show how the intermediate thresholds A, B are not evenly spaced relative to 0, 255: their spacing too contributes to the linearization process. Also evident is that the linearization curve is the main contributor in lower-tone regions (0 to A), whereas it is practically a straight line as the different thresholds approach more closely (A to B, B to 255). Therefore when the system halftones at 25 dots/mm at four bits, most of the linearization work can be done through the threshold definition.

(c) <u>Linearization and threshold examples</u> — Finally, the result for a real case in a preferred embodiment (with drop table of [0 1 1 2] at 25 dots/mm, two bits) will be helpful for clearer understanding (Fig. 12). This represents a current Hewlett Packard product.

8. INK LIMITING AND PLANE SPLIT

(a) <u>Overview</u> — Based on the foregoing understandings of how ED works, the next step upstream in Fig. 5 is to consider feeding of data into the ED. This system is using plane-independent error diffusion — meaning that no consideration is made, when deciding about one color, of decisions already made for other colors.

In the product which is a preferred embodiment, error-diffusion processing proceeds alternatively left to right and then right to left along consecutive rows. The printheads are six in number — KCMYcm — while the input files are always KCMY (once they have gone through the

SUBSTITUTE SHEET 6/14/5

color pipeline, which may transform them from RGB to KCMY).

In design of this system there were several choices concerning the ideal point at which to split the cyan and magenta planes between dark and light inks. It was decided to split before halftoning, and thus to pass six independent planes of data into the error diffusion stage.

The split between dark and light inks is not trivial, in particular because there are different combinations of dark and light ink delivering the same color, but not the same total amount of ink. In other words, the plane-split process must be ink-dependent.

Therefore, it is a good point at which to perform ink limiting. The main disadvantage of this process is that it operates at pixel level, not object level.

In other words, if there is a large solid area of the same color, the system must still repeat the same operation for each pixel, even though it must always yield the same result. This feature compels design of an algorithm that gives a good tradeoff between image quality and throughput.

(b) Depletion algorithm — We may distinguish three stages in the ILPS (ink-limiting and plane-split) process (Figs. 13 and 14). First, it is necessary to determine how much ink is to be fired onto the particular pixel being processed.

Because of all the configurable parameters throughout the halftoning pipeline (linearization, thresholds, superpixel families, and drop table), it would be impossible to predict the ink usage based on only the values of the input image. Therefore for each channel a lookup table (LUT) must be built to associate the channel value to the ink usage.

SUBSTITUTE SHEET 6/14/5